

ZX Fab Standard Manager 0.6.0.0-r1 使用說明(白話中文版)

Reality Matrix AI Inc. · 2026-09-24

適用版本:**0.6.0.0-r1 pre-beta**(試用預覽版)

上架時刻:**2026-09-24 17:51**(台北時間)· 下載位置:官網首頁 <https://www.zxbuilder.club/> 的 Downloads 區塊· 壓縮包指紋見第一節 1.2(本公司於上架後自官網下載同一顆親自計算)

本說明引用的按鈕與訊息文字依實際介面書寫,實際視窗與按鈕文字以畫面為準。已經看過 0.5.0.0-r1 版說明的人:0.6.0.0 與 0.6.0.0-r1 這兩版新增了什麼,直接看第四節與第七節。

○、這是什麼

ZX Fab Standard Manager(以下簡稱「標準管理台」)是一套 Windows 桌面軟體,做一件事:把廠房的「資訊標準」變成一份可以存檔、可以對照、可以交出去的檔案。

所謂資訊標準,就是這些規定:廠房裡有哪些類別的設備(管線、風管、電氣設備.....)、每個類別下有哪些族群與類型、每個設備要填哪些參數、用什麼單位、哪些一定要填。標準管理台用 Revit 使用者熟悉的語言(類別 → 族群 → 類型 → 參數)把這些規定整理成一份專案檔(.zxfsm),並且可以:

- 匯出成 Revit 可以直接載入的共用參數檔(.txt)。
- 匯出成標準包(.zxf),交給 **ZXBuilder**(裝在 Revit 裡的外掛)使用 —— ZXBuilder 匯出 ZF USDC 零件庫與 ZX USDC 場景時,會拿這份標準包來對照與檢查。
- 讀入 ZXBuilder 匯出的 ZX USDC 場景檔或 ZF USDC 零件庫檔,把裡面的元件對到你的標準,並在 3D 視窗檢視。
- 把同一份標準對到 PI Asset Framework(以下簡稱 PI AF)的樣板、屬性、階層與點位,匯出成 Excel,或直接套用到 PI AF 伺服器。
- 本版起:在零件類型上定義 **IoT 點位**(不必先綁 Revit 參數),並把它們一起放進標準包。

兩邊的分工

誰	負責
ZXBuilder (裝在 Revit 裡)	從 Revit 匯出零件庫檔(ZF USDC)與場景檔(ZX USDC);把場景匯回 Revit;讀取標準包
標準管理台(本軟體)	制定與維護資訊標準;讀 ZF USDC / ZX USDC 來對照標準;把標準包與共用參數檔交給 ZXBuilder 與 Revit;維護 PI AF 對照表與 IoT 點位

本軟體不編輯 3D 幾何、不寫回 USD 檔、不改 Revit 模型。要改模型請在 Revit 裡改,再用 ZXBuilder 重新匯出。

一、取得與安裝

1.1 系統需求

項目	需求
作業系統	64 位元 Windows
執行環境	不需要另外安裝任何執行環境,壓縮包已經包含所有需要的元件
磁碟空間	解壓後 869 個檔案,共 200,831,027 位元組(約 201 MB;Windows 檔案總管顯示約 191 MB)
顯示卡	3D 檢視視窗需要支援 OpenGL 4.5 Core 的顯示卡與驅動程式。不支援時程式不會當掉,3D 視窗會顯示「3D 顯示初始化失敗」並跳一個訊息框,主視窗其餘功能照常使用
Excel 匯出	不需要在電腦上安裝 Excel,活頁簿由程式自己產生
網路	只有「連線設定」與「套用到 AF」需要連到你的 PI AF 伺服器;其餘所有功能都不需要網路

其餘硬體下限(Windows 版本、記憶體容量)我們尚未實際測過,因此不列數字。若在你的機器上無法啟動,請與我們聯絡。

1.2 取得壓縮包

- 本軟體解壓即執行,沒有安裝精靈,也沒有解除安裝程式(要移除就把解壓出來的資料夾刪掉)。
- 取得方式:官網首頁 <https://www.zxbuilder.club/> 的 Downloads 區塊,「配套工具 — ZXFabStandardManager」項目,按下載鈕。
- 壓縮包資訊:檔名 `ZXFabStandardManager-0.6.0.0-r1-win-x64.zip` · 檔案大小 82,268,082 位元組 · SHA-256 `B52C99C7426919AAB101EBB02330CC0B46C0CE3B6A26AF0664225FE4B3DA35B` · Windows x64,自帶執行環境。

1.3 核對壓縮包指紋(建議下載後做一次)

任選一種做法,在壓縮包所在資料夾執行:

```
Get-FileHash .\ZXFabStandardManager-0.6.0.0-r1-win-x64.zip -Algorithm SHA256
```

或

```
certutil -hashfile ZXFabStandardManager-0.6.0.0-r1-win-x64.zip SHA256
```

把印出來的那一長串英數字和上面 1.2 的 SHA-256 逐字比對。一字不差才解壓使用;對不上請重新下載,不要使用。

1.4 第一次啟動會看到什麼

- 1. 把壓縮包解壓到任一個你有寫入權限的資料夾(例如 D:\ZXFabStandard\)。
- 2. 進入解壓後的資料夾,執行 ZXFabStandardManager.exe 。
- 3. 主視窗打開,標題列顯示 ZXFabStandardManager v0.6.0.0 pre-beta ;開啟專案後,標題列後面會再接上專案檔名。⚠ 標題列與「關於」視窗顯示的是四段版號 v0.6.0.0 ,不含 -r1 ;要確認手上有 r1 修正版,看壓縮包檔名與 1.2 的 SHA-256,或在檔案總管對 ZXFabStandardManager.exe 按右鍵 → 內容 → 詳細資料,「產品版本」欄開頭為 0.6.0.0-pre-beta-r1 。
- 4. 第一次啟動時大部分按鈕是灰的,只有「新建」「開啟」「檢閱警告」「關於」「回報問題」這幾顆可以按;「最近使用的檔案」下拉要開過檔案之後才會亮 —— 這是正常的,建立或開啟一份專案後其他按鈕就會亮起。
- 5. 在一個全新的資料夾第一次啟動,不會跳任何對話框。

啟動時可能遇到的兩件事

- 只能開一份:已經有一份在跑時,再開一次不會開出第二份視窗;新開的那一次會把啟動參數交給已經開著的那一份,然後自己結束,已開的視窗會跳到最前面。
- 缺元件的錯誤框:解壓不完整時,啟動會跳出標題含 DLL_MISSING 的訊息框,內文第一行「必要元件缺失或無法載入。」,接著列出缺的是哪個檔案、原因與路徑,最後一行「請重新安裝 ZXFabStandardManager。」;按確定後程式結束。處理方式:重新解壓整包,不要只複製執行檔;回報問題時請把檔案 / 原因 / 路徑三行一起附給我們。

二、畫面總覽

2.1 主視窗

主畫面分三層:

標準	參數	檢查	匯出	PI AF	說明	← 功能分頁(上)
[按鈕]	[按鈕]	[按鈕]	...			

專案瀏覽器	工作區	性質
(左,樹狀)	(中)	(右,清單)

就緒	上次落盤 HH:mm:ss	已授權	v0.6.0.0 pre-beta · build ...	← 狀態列(下)
----	---------------	-----	-------------------------------	----------

區域	用途
功能分頁	六個分頁:標準、參數、檢查、匯出、PI AF、說明。每個分頁下面是一排按鈕
專案瀏覽器(左)	樹狀顯示你的標準:最上層是標準本身,往下是類別 → 族群 → 類型。點選一個節點,右邊的性質面板顯示它的內容
工作區(中)	平常上方顯示一行灰底提示「驗證 > ZX USDC / ZF USDC 載入場景。」,下面是空的。切到「檢查」分頁並載入場景檔後,這裡會變成場景元件的樹狀清單

區域	用途
性質 (右)	顯示你在左邊選到的節點有哪些參數、值是多少、從哪裡繼承來的。選到「類型」時,上方多一顆「類型性質...」按鈕
狀態列 (下)	左邊是最近一次操作的結果(例如「已儲存·檔名」「PI AF 對映表已存為版本 2(3 項變更)」);右邊依序是「上次落盤 HH:mm:ss」、授權狀態、版本、建置時間與介面語言代碼。⚠️「上次落盤」是你每改一筆它就更新,不是按了「儲存」才動

各分頁的按鈕(繁體中文介面,括號內為英文介面):

分頁	按鈕
標準 (Standard)	新建(New)· 開啟(Open)· 最近使用的檔案:(Recent Files, 下拉選單)· 儲存(Save)· 品類(Category)· 族群(Family)· 類型(Type)· 載入種子(Load Seed)· 關聯(Relationships)· 系統類型(System Types)
參數 (Parameters)	共用參數(Shared Parameters)· 專案參數(Project Parameters)· 專案單位(Project Units)
檢查 (Validate)	檢查(Validate)· 檢閱警告(Review Warnings)· ZX USDC · ZF USDC · 對映(Mapping)
匯出 (Export)	共用參數檔(Shared Parameters File)· 標準包 (.zxf)(Standard Package (.zxf))· Excel · PI AF Excel · 送到 ZXBuilder(Send to ZXBuilder)
PI AF	Template 對映(Template Mapping)· Attribute 對映(Attribute Mapping)· 樹階層規則(Hierarchy Rules)· Point 規則(Point Rules)· 連線設定(Connection)· 套用到 AF(Publish to AF)
說明(Help)	關於(About)· 回報問題:開啟資料夾 · 回報問題:複製路徑

2.2 3D 檢視視窗

「檢查 → ZX USDC」或「檢查 → ZF USDC」載入檔案時,程式會另開一個深色的獨立視窗(下文簡稱「3D 視窗」),標題列顯示 `ZXFabStandardManager v0.6.0.0 · built <建置時間>`。它預設以 1280 × 800 的一般視窗置中開啟,不會自動最大化;它是主視窗的子視窗,會顯示在主視窗前面(可以自行拖開或縮小),關閉主視窗時會一起關閉。

- 滑鼠左鍵點一個元件,它會被紫框框起來;視窗下方的選取狀態列顯示 **Selected:** [編號] 元件名。
- 右上角工具列:  /  鏡頭後退 / 前進(鍵盤 S / W)、 Floor ON 開關地板、 燈光亮度(旁邊顯示目前倍率,例如 1.00)、 Fit All 讓整個場景填滿畫面(鍵盤 F)、Opacity: 透明度。
- 功能表(繁體中文介面,括號內為英文介面):檔案(File):開啟 USD...(Open USD...)、加入 USDC 至場景...(Add USDC to Stage...)、清空場景(Clear Stage)、結束(Exit);檢視(View):全部符合視窗 (F)(Fit All (F))、重置 (R)(Reset (R));模型(Models);樓層(Levels);說明(Help):關於(About)。
- 本版新增:「IoT 點位」側面板——第一次點選任何元件之後,畫面右上角會浮出一塊 260 像素寬的深色面板,標題「IoT 點位」,用來替你點到的那個類型新增 IoT 點位。它疊在 3D 畫面上方,不會擠壓 3D 畫面。詳見 4.1。
- 這個視窗的少數文字仍是英文(工具列的 Floor ON / Fit All 兩顆按鈕、透明度欄位、選取狀態列、面板裡三個後端系統名稱)。

2.3 「關聯」視窗

「標準 → 關聯」打開,視窗標題「關聯」。它把整份標準畫成一張可以拖曳、縮放、點選的關係圖。0.6.0.0 起這個視窗多了搜尋、深度、佈局三組控制項與三種新節點,詳見 4.3。

2.4 側邊視窗(名稱與打開方式)

視窗	從哪裡開
共用參數	參數 → 共用參數
專案參數	參數 → 專案參數
專案單位	參數 → 專案單位
系統類型	標準 → 系統類型
對映(場景元件 ↔ 標準)	檢查 → 對映(載入場景後才亮)
PI AF · 對映(四頁:Template 對映 / Attribute 對映 / 樹階層規則 / Point 規則)	PI AF 分頁前四顆按鈕,各開到對應那一頁
連線設定	PI AF → 連線設定
套用到 AF	PI AF → 套用到 AF
關於	說明 → 關於(看版本、貼授權碼、切換介面語言:English / 中文 / 日本語 / 한국어)

三、基本流程

下面是最常見的一條路,從零到 ZXBuilder 讀到你的標準包。每一步都寫「你做什麼」和「你會看到什麼」。

3.1 建立或開啟標準包專案

步	你做什麼	你會看到
1	執行 <code>ZXFabStandardManager.exe</code>	主視窗,大部分按鈕是灰的
2	(可選)說明 → 關於 → 右上角選「中文」	整個介面變中文而且會記住
3	標準 → 新建 → 選資料夾與檔名(例如 <code>FabStandard-2026.zxfsm</code>) → 存檔,或標準 → 開啟 選既有的 <code>.zxfsm</code>	狀態列「已開啟 · 檔名」;左邊樹出現根節點;按鈕都亮了
4	標準 → 載入種子	左邊樹長出 40 個常用 Revit 類別、系統代碼、常用參數與單位;狀態列顯示「已載入種子 · +新增筆數 (skipped 略過筆數)」(略過筆數用英文 skipped 標示)。按第二次不會重複灌

開啟舊版程式存過的專案檔時會自動升級格式,狀態列會註明升版。反過來,本版存過的專案檔,拿回較舊版的程式開會看到 `PROJECT_VERSION_NEWER`,請整個團隊一起更新。

3.2 建立類別 / 族群 / 類型(等於「匯入」你的零件分類)

步	你做什麼	你會看到
5	左邊樹點根節點 → 標準 → 品類 → 輸入名稱 → 確定	多一個類別
6	左邊樹點剛建的類別 → 標準 → 族群 → 輸入名稱 → 確定	該類別下多一個族群
7	左邊樹點剛建的族群 → 標準 → 類型 → 輸入名稱 → 確定	該族群下多一個類型

步驟 6、7 一定要先在左邊樹點選父節點再按按鈕,否則會跳「請先在專案瀏覽器選取類別或族群。」。

如果你已經有 ZXBuilder 匯出的 ZF USDC 零件庫檔或 ZX USDC 場景檔,也可以反過來走:「檢查 → ZF USDC / ZX USDC」載入後,用「檢查 → 對映」把裡面的品項對到你的類別 / 族群 / 類型,對好按確定,狀態列「對映已存回·筆數·專案檔名」。下次載入同一個檔會自動套用。

3.3 設定參數與屬性

步	你做什麼	你會看到
8	參數 → 共用參數 → 新建... → 填名稱、規程、參數類型、單位 → 按「新 GUID」→ 確定	清單多一列參數。要匯出給 Revit 的參數一定要有 GUID
9	同一視窗選那個參數 → 綁定... → 選族群、勾必填、選階段 → 確定	下方允許矩陣對應格子出現 ✓(必填)或 ○(有綁但非必填)
10	(可選)左邊樹點某個類型 → 右邊「類型性質...」→ 改值 → 確定	狀態列「類型性質已存回(N 項變更)」
11	(可選)標準 → 系統類型 → 新增或編輯 → 儲存	有問題的列會變粉紅底加紅框,全部通過才一次寫入
12	檢查 → 檢查 → 檢查 → 檢閱警告	警告清單,點一列就跳到現場;完全沒問題時顯示「0 個警告」

3.4 PI AF 對映(把設計語言接到營運語言)

步	你做什麼	你會看到
13	PI AF → Template 對映	「PI AF·對映」視窗開在第 1 頁;左欄是你的品類 / 族群 / 類型(唯讀),右欄可編
14	在右欄填每一列的 Template 名(留空則自動用前綴名);用不到的列勾「排除」	右欄即時更新
15	切到 Attribute 對映,上方選一列 Template ;在左欄選要對映的參數,按「新增 →」	右欄多一列,再填 Attribute 名、值型別、UOM
16	(本版新增)要加一個不綁 Revit 參數的 IoT 點位;按「新增 IoT 點位」	右欄多一列,「參數」欄空白;填屬性名與點位欄。詳見 4.1
17	切到 樹階層規則 ,按「新增」加層,至少一個視角為 Main ,層序從 1 開始連續	每按一次多一列
18	按「檢查」	驗證列顯示六個數字,全過是綠字,有問題是紅字加原因碼

步	你做什麼	你會看到
1 9	按「確定」	存回專案;內容和上一版有差就自動存成新的一版,狀態列「PI AF 對映表已存為版本 N(M 項變更)」;沒差則「PI AF 對映表:無變更(版本 N)」

3.5 匯出標準包

步	你做什麼	你會看到
20	標準 → 儲存	狀態列「已儲存 · 檔名」
21	匯出 → 送到 ZXBuilder	對話框列出標準包 .zxf 、共用參數檔 .txt 與交換資料夾三個路徑,有「開啟資料夾」按鈕;狀態列「已送到 ZXBuilder · 檔名」
21- 另	或:匯出 → 標準包 (.zxf) 自己選位置存	狀態列「已匯出標準包(N 表 · M 列) · 檔名」

標準包是加密格式,用記事本打開只會看到亂碼,這是正常的。裡面有你的標準本體、整套 PI AF 對照表、本版起還有每個類型的 **IoT** 點位與每個樣板類別的 Revit 內建類別名(見 4.2),加上內容指紋;你的 PI AF 連線設定與套用結果不會被放進去。

3.6 在 **ZXBuilder 5.8.0.0** 讀取

按過「送到 ZXBuilder」之後,兩個檔的路徑已寫進系統設定,ZXBuilder 下次匯入時會自動讀到標準本體,不用再手動指定路徑。本版產出的標準包可以被 **ZXBuilder 5.8.0.0** 讀取;版本相容細節見第六節。標準管理台這一側不會去檢查你的 ZXBuilder 是哪一版,版本要你自己顧。

四、0.6.0.0 / 0.6.0.0-r1 新功能逐步

4.1 IoT 點位免綁 Revit 參數(0.6.0.0-r1 新增)

這是什麼:以前 PI AF 對照表的每一個 Attribute 都必須對到一個 Revit 參數,沒有參數就放不進去。但很多 IoT 量測點(溫度、流量、狀態)在 Revit 模型裡根本沒有對應參數。本版起,你可以直接在一個零件類型上定義 IoT 點位,不必先在 Revit 建參數;點位會跟著標準包一起匯出。

每個點位有哪些欄位(兩個入口的欄位相同):

欄位(畫面文字)	意思	怎麼填
屬性名 / Attribute 名(Attribute name;3D 面板寫「屬性名」,對映視窗的欄名寫「Attribute 名」,是同一個東西)	這個點位在 PI AF 上的 Attribute 名稱	一定要填;同一個 Template 裡不能有兩個同名
值型別(Value type)	值的資料型別	下拉五選一: Double · Int32 · String · Boolean · DateTime ,預設 Double
單位(Unit)	工程單位	可留空;填了就必须是你標準裡「專案單位」表裡有的單位名(在對映視窗的 UOM 欄也可以直接打單位符號,程式會換成單位名)

欄位(畫面文字)	意思	怎麼填
點位種類(Point Kind)	這是固定屬性還是量測點	下拉:空白・ static (固定值,例如額定功率)・ measure (隨時間變的量測點,例如目前流量);兩個入口新增時都預設 measure
點位名(Point Name)	這個點位在後端系統裡的通用名稱	可留空;留空時讀取端沿用既有規則「元件名.屬性名」
AVEVA PI AF	在 AVEVA PI AF 裡的名稱	可留空;留空時退回「點位名」
TDengine	在 TDengine 裡的名稱	同上
AVEVA Historian	在 AVEVA Historian 裡的名稱	同上

入口一:PI AF 對映視窗

步	你做什麼	你會看到
1	開啟專案 → PI AF → Attribute 對映	「PI AF・對映」視窗開在第 2 頁
2	上方下拉選一系列 Template (這個點位要掛在哪個樣板下)	左欄列出該樣板綁定的參數;右欄是已有的 Attribute 列
3	按右側第三顆按鈕「新增 IoT 點位」(前兩顆是「新增 →」「刪除」)	右欄末尾多一系列並自動選中、捲到看得見的位置;它的「參數」欄是空白的(因為沒綁參數),「點位種類」預設 measure 。沒選 Template 就按,什麼都不會發生
4	在那一系列填「 Attribute 名」,選值型別、UOM;往右捲到最後五欄填「點位種類」「點位名」「AVEVA PI AF」「TDengine」「AVEVA Historian」	右欄即時更新
5	按「檢查」	驗證列全綠;或紅字列出原因碼(見下)
6	按「確定」	存成新的一版;主視窗狀態列「PI AF 對映表已存為版本 N(M 項變更)」

入口二:3D 檢視視窗選件側面板

前提:專案已開啟、該類型已經在標準裡(類別 → 族群 → 類型都對得到)、而且該類型在 **PI AF** 對映表最新版裡已經有一列 **Template**(在 **Template** 對映頁填過)。

步	你做什麼	你會看到
1	檢查 → ZX USDC → 選 ZXBuilder 匯出的場景檔	3D 視窗開啟,主視窗中間長出元件樹
2	在 3D 視窗用滑鼠左鍵點一個元件	元件被紫框框起;右上角浮出「IoT 點位」面板,第二行顯示它對到的「類別、族群、類型」
3	看面板最下方那一行黃字提示	沒有黃字、「新增 IoT 點位」按鈕可按 = 條件都符合;有黃字 = 還不能加,黃字會直接告訴你為什麼(五種訊息見下)

步	你做什麼	你會看到
4	填屬性名、值型別、單位、點位種類、點位名與三個後端名稱 → 按「新增 IoT 點位」	成功:主視窗狀態列「PI AF 對映表已存為版本 N(1 項變更)」,面板的文字欄清空(值型別與點位種類保留);失敗:主視窗狀態列顯示原因碼(例如 ATTR_NAME_EMPTY),面板內容保留讓你改

⚠ 從 3D 面板新增的點位每按一次就直接存成新的一版,不用再開對映視窗按確定。

3D 面板五種黃字提示

你會看到	原因	怎麼做
未選取元件。	還沒點到任何元件	在 3D 畫面點一個元件
請先開啟標準。	主視窗還沒開專案	標準 → 開啟 / 新建
此類型不在標準中。	點到的元件對不到你標準裡的「類型」層(只對到族群或類別,或完全對不到)	先在標準裡建好該類型,或用「檢查 → 對映」把它對上
請先關閉 PI AF 對映視窗。	「PI AF · 對映」視窗還開著	關掉它再按
此類型尚無樣板,請先於 PI AF 對映視窗建立樣板。	該類型在對映表最新版裡沒有 Template 列	PI AF → Template 對映 → 替該類型那一列填 Template 名 → 確定

填錯會看到的原因碼(兩個入口共用同一套檢查)

原因碼	意思	怎麼做
ATTR_NAME_EMPTY	屬性名沒填	填屬性名
ATTR_NAME_DUP	同一個 Template 裡有兩個同名的 Attribute	換一個名稱
UOM_INVALID	單位不在你標準的單位表裡	從下拉選,或先到「參數 → 專案單位」確認單位名
PIPOINT_NO_PATTERN	Data Reference 選了 pipoint 卻沒填 Config String	填樣式,或把 Data Reference 改回 static

匯出後在標準包裡長什麼樣:標準包的 PI AF 節裡,每個 Attribute 是一筆紀錄。免綁參數的點位,revit_parameter 欄是空字串、guid 是空值,並多出三個鍵(kind、point_name、backend_names)。下面是一筆的樣子(鍵名照標準包實際寫出的順序):

```
{
  "af_template": "...",
  "revit_parameter": "",
  "guid": null,
  "af_attribute": "屬性名",
  "value_type": "Double",
  "uom": "...",
  "data_ref": "static",
  "config_string": null,
  "is_config_item": false,
  "kind": "measure",
  "point_name": "點位名",
```

```
"backend_names": { "aveva_pi": "...", "tdengine": "...", "historian": "..."}
}
```

「點位種類」留空白時不會寫出 `kind` 鍵;三個後端名稱都空白時不會寫出 `backend_names`。PI AF Excel 匯出的 `AttributeTemplates` 工作表本版不含這五個新欄(點位種類、點位名、三個後端名稱);IoT 點位那幾列仍會列在這張表裡,但 `Description` 欄是空白的(因為沒有 `Revit` 參數),從 Excel 看不出點位種類與後端名稱。要交給第三方時請以標準包或畫面為準。

4.2 樣板類別帶 Revit 內建類別名(0.6.0.0-r1 新增)

這是什麼:標準包裡每一列 `Template` 原本只寫類別的顯示名(例如 `Mechanical Equipment`)。「載入種子」灌進來的類別名一律是英文;中文介面下左邊的樹雖然會把它顯示成「機械設備」,寫進專案檔與標準包的仍是英文名。不同語系的 `Revit` 對同一個類別顯示的名字不一樣(中文 `Revit` 顯示的是「機械設備」),ZXBuilder 拿顯示名去比對就可能對不上。本版起,每一列 `Template` 多寫一個 `category_ost` 欄,值是 `Revit` 內部的內建類別名(例如 `OST_MechanicalEquipment`),這個名字在任何語系的 `Revit` 裡都一樣。

你要做什麼:什麼都不用做。匯出標準包(不論走「匯出 → 標準包 (.zxf)」還是「送到 ZXBuilder」)時程式自動填入;顯示名欄照舊保留。

哪些類別會有值:「載入種子」灌進來的 40 個 `Revit` 類別都有。你自己用「品類」按鈕新建、名字不在那 40 個裡的類別,這一欄會是空字串,讀取端會退回用顯示名比對。

4.3 0.6.0.0 新增:「關聯」視窗的互動功能

0.6.0.0(2026-09-24 上午上架)把「標準 → 關聯」從一張靜態的圖變成可以互動的圖。0.6.0.0-r1 沿用,沒有再改。

視窗上方多了什麼(從左到右一排,窄視窗時會自動換行):

控制項	做什麼
八個核取方塊	依節點種類顯示 / 隱藏:類別 · 族群 · 類型 · 系統類型 · 零件 · PI AF 對映版本 · PI AF 樣板 · PI AF 屬性(後三個是新的)
載入 identity.csv ...	載入零件識別對照檔,圖上多一層零件節點
佈局 (Layout)	下拉五選一:高效分層 (EfficientSugiyama) · 分層 (Sugiyama) · 力導向 (FR)(預設) · 限界力導向 (BoundedFR) · 彈簧 (KK)。括號裡是演算法代號,照畫面文字寫。換佈局不會掉節點
搜尋 (Search)	打字即過濾:名稱含該字串(不分大小寫)的節點保持清楚,其餘淡化;右邊顯示「命中 N」。按 Enter 跳到第一個命中並用橘框框起;按 Esc 清空搜尋並全部還原
深度 (Depth)	下拉:全圖 · 0 · 1 · 2 · 3。先點一個節點,再選深度,就只畫該節點往外 N 層內的節點與邊,旁邊顯示「N 節點 / M 邊 / 深度 d」;選回「全圖」還原

圖上多了什麼

節點	代表	樣子
PI AF 樣板	對映表最新版裡沒勾「排除」的 Template 列	深紅色、直角
PI AF 屬性	該樣板底下的 Attribute 列	棕色、對角圓角
PI AF 對映版本	對照表的每一個版本 (v1、v2 ...)	青綠色、上方圓角;相鄰版本之間用一條灰色粗實線串起來,自成一群
(既有)類別 / 族群 / 類型	—	外框多了顏色:綠框 = 這個節點在對映表最新版裡有 Template;灰框 + 半透明 = 沒有。有 Template 的節點右上角還有一個 4 × 4 的小方塊:藍色 = Template 名是你自己填的,橘色 = 你留空、程式用自動前綴名

連線:對映邊(類別 / 族群 / 類型 → PI AF 樣板)與樣板 → 屬性的隸屬邊都是深紅色實線。

點選節點會發生什麼:點一個節點,它和它的一階鄰居保持清楚,其餘節點與邊降到 25% 不透明度;類別 / 族群 / 類型三種節點會同時讓主視窗左邊的專案瀏覽器選中同一個節點。按 **Esc** 或點圖上空白處還原。如果搜尋框裡有字,第一次 Esc 先清搜尋,再按一次才清高亮。

上方的統計列:「N 個類別 · N 個族群 · N 個類型 · N 個系統類型 · N 個關聯 · 顯示 N / N」;有 PI AF 樣板時後面再接「· PI AF 樣板 N · 屬性 N」。

對不上的東西不會被安靜丟掉:圖上方會用紅字列出這次載入時被擋下的項目(例如某條關聯的端點找不到、某個樣板指向不存在的節點),每一類最多列 20 行。

五、常見問題與錯誤訊息

每一條 = 你會看到什麼字 → 原因 → 怎麼做。

你會看到	原因	怎麼做
訊息框 DLL_MISSING / DLL_LOAD_FAILED, 程式結束	解壓不完整,缺必要元件	重新解壓整個壓縮包
MESHCLI_MISSING 「USD 載入元件缺失,請重新安裝 ZXFabStandardManager。」	缺 USD 讀取元件	重新解壓整個壓縮包
NOT_USDC 「此檔不是 USDC 檔。」	你選到的不是 USD Crate 格式	選 ZXBuilder 匯出的 .usdc
PROJECT_VERSION_NEWER	專案檔是更新版程式建的	依訊息指示更新程式;本版存過的檔拿回舊版開就會這樣
PROJECT_READONLY	檔案是唯讀屬性;或它是舊版格式而你選了唯讀開啟	取消唯讀,或改用一般方式開啟讓它升版
PROJECT_LOCKED / PROJECT_BUSY	檔案被另一個程式佔用 / 等超過三秒仍拿不到	關掉另一邊再開

你會看到	原因	怎麼做
PROJECT_NOT_LOCAL	檔案在雲端同步夾但還沒下載到本機	先讓它下載完整再開
「請先在專案瀏覽器選取類別或族群。」 (SELECT_PARENT)	按「族群」或「類型」前沒先點父節點	左邊樹先點類別或族群
NO_PROJECT 「請先開啟或新建標準專案,才能存回對映。」	沒開專案就要存回對映	先新建或開啟專案
3D 面板黃字「此類型尚無樣板,請先於 PI AF 對映視窗建立樣板。」	該類型在對映表最新版裡沒有 Template 列	見 4.1 入口二
3D 面板黃字「此類型不在標準中。」	點到的元件只對到族群或類別,沒對到類型	先建好類型或用「對映」對上
狀態列 ATTR_NAME_EMPTY / ATTR_NAME_DUP / UOM_INVALID	IoT 點位或 Attribute 的欄位沒過檢查	見 4.1 原因碼表
對映視窗驗證列「存檔被拒」	按確定時有任何一條沒過	照紅字原因碼補完再按確定;你的東西不會遺失
對映視窗 TEMPLATE_NAME_DUP / BASE_MISSING	同一版兩列用了同一個 Template 名 / 指定的 Base 找不到	改名 / 改選存在的 Base
對映視窗 MAIN_VIEW_MISSING / LEVEL_GAP	樹階層沒有 Main 視角 / 層序不連續	加一列 Main,層序從 1 開始
關聯視窗紅字 GRAPH_EMPTY	標準裡還沒有任何類別 / 族群 / 類型 / 系統類型	先建標準
關聯視窗紅字 GRAPH_LAYOUT_FAILED	版面計算失敗	換一種佈局再試;仍失敗請依本節末「遇到問題要怎麼回報」把追蹤檔寄給我們
3D 視窗「3D 顯示初始化失敗」	顯示卡或驅動不支援 OpenGL 4.5 Core	更新顯示卡驅動後重試;主視窗其餘功能照常
狀態列紅字「未授權·唯讀示範」;寫入時 READONLY_DEMO	這台電腦還沒啟用授權	說明 → 關於 → 「機器碼」旁按「複製」,把機器碼寄給我們 → 收到授權碼後在「授權碼」欄按「貼上並啟動」(授權與 ZXBuilder 同一張)
EXPORT_FAILED	目標資料夾不可寫入	換資料夾;原本的檔案不會被動到
SEND_FAILED	交換資料夾寫不進去,或送出前的格式自我檢查沒過	確認資料夾可寫;先按「檢查 → 檢查」補完問題再送

問:在 **ZXBuilder** 那邊看不到 **PI AF** 對照表或 **IoT** 點位? 先確認 **ZXBuilder** 版本(見第六節)。標準管理台這一側不會報錯,版本不夠時 **ZXBuilder** 只是讀不到那一段。

問:標準包用記事本打開是亂碼? 正常,標準包是加密格式。

問:兩份標準包的內容指紋不一樣,是不是有人改過? 同一版程式各匯一次再比;不同版程式匯出的指紋本來就不能互比。

遇到問題要怎麼回報:記下畫面上的大寫英文碼 → 說明 → 回報問題:開啟資料夾 → 找最新的 `.zip` 追蹤檔 (檔名形如 `ZXFSM_20260924_175100_123.zip`,是日期加時間,最新的排最後)→ 連同操作步驟寄到 `michael.lin@realitymatrixai.com`。追蹤檔是加密的,裡面只有操作紀錄與檔名,沒有你的模型幾何與參數值。

六、相容性

項目	說明
本版標準包 ↔ ZXBuilder 5.8.0.0	0.6.0.0-r1 匯出的標準包可以被 ZXBuilder 5.8.0.0 讀取。標準包裡多出來的 IoT 點位欄位與 <code>category_ost</code> 欄,對不認得它們的讀取端是「多出來的鍵」,依標準包格式規定會被忽略,不會造成載入失敗
要讓 ZXBuilder 用到新欄位	IoT 點位欄位與 <code>category_ost</code> 是配合 ZXBuilder 5.8.0.0-r1 (2026-09-24 上架)設計的;要在 ZXBuilder 端用到這兩項,請更新到該版
舊的標準包	用 0.5.0.0-r1 或 0.6.0.0 匯出的標準包不必重出;讀取端遇到沒有 <code>category_ost</code> 的舊包會退回用顯示名比對
專案檔 <code>.zxfsm</code>	本版會把舊版專案檔自動升級;升級後的檔案舊版程式打不開(<code>PROJECT_VERSION_NEWER</code>)
PI AF 對照表 (0.5.0.0 起)	讀取端需 ZXBuilder 5.8.0.0 以上;較舊的 ZXBuilder 照常載入標準本體,但會安靜略過對照表
加密標準包 (0.3.0.0 起)	讀取端需 ZXBuilder 5.7.9.3-r2 以上
授權與介面語言	與 ZXBuilder 共用同一張授權、同一個介面語言設定;在任一邊改了,另一邊下次啟動時就跟著變(兩邊都是啟動時才讀這個設定)

七、版本資訊

版本	上架時刻 (台北)	客戶看得到的變化
0.6.0.0	2026-09-24 08:53	「關聯」視窗改為互動式:點選高亮鄰居、深度 0-3 的局部圖、PI AF 樣板 / 屬性 / 對映版本三種新節點、已對映 / 未對映外框與來源角標、搜尋框、佈局下拉(預設力導向)。詳見 4.3
0.6.0.0-r1	2026-09-24 17:51	① IoT 點位可以不綁 Revit 參數,從「PI AF 對映視窗 → 新增 IoT 點位」或「3D 視窗選件側面板」建立,帶點位種類、點位名與 AVEVA PI AF / TDengine / AVEVA Historian 三個後端名稱,並隨標準包匯出(4.1);② 標準包每列 Template 多寫 Revit 內建類別名 <code>category_ost</code> ,讓 ZXBuilder 在任何語系的 Revit 都對得上(4.2)。標準包仍可被 ZXBuilder 5.8.0.0 讀取

本版沒有的(與 0.5.0.0-r1 相同,列出常被問到的):依檢查結果在 3D 畫面上為元件上色、輸出檢查報告檔;PI AF 對照表的逐欄差異檢視畫面;在 PI 上建立 PI Point 或把值寫進 PI(「套用到 AF」只建 Element、綁 Template、寫入固定值屬性);從 AF 反向匯入;PI AF Excel 的 `AttributeTemplates` 工作表不含 IoT 點位的五個新欄;安裝程式與解除安裝程式。

幾件請你知道的事:PI AF 的連線與套用是在模擬伺服器上完成驗證的,尚未在真實的 PI AF 伺服器環境驗證過,第一次在貴司環境使用前建議先與我們聯繫安排陪同測試,並先在測試用的 Database 上試跑;「關聯」視

窗沒有節點數量上限,標準規模很大時開啟可能需要一段時間。

聯絡:michael.lin@realitymatrixai.com

© 2026 Reality Matrix AI Inc. · ZX Fab Standard Manager 為 ZXBuilder 產品系列之一 · michael.lin@realitymatrixai.com